AMARA Algorithm

Andrzej Szablewski

November 23, 2020

1 Introduction

The AMARA algorithm is an asymmetric (public key/private key) encryption method based on binary matrices developed in 2016 by Andrzej Szablewski, Radosław Peszkowski and Mateusz Janus under supervision of Prof. Tomasz Szemberg. The idea for creating such a method originated from the research on specific sets permutations [1].

2 Algorithm overview

Both public and private keys are considerably large binary matrices of a size n x n, which are inverse to each other. Data to encrypt or decrypt must be in the binary format. Cipher breaking difficulty lies in the matrix inversion complexity, which lower bound has been proven [2] to be:

$$O(n^2\ log(n)),$$

which is not an impressive complexity comparing to currently used methods. However, since AMARA encryption and decryption processes are based on *xor* operation between certain elements of the matrix, the algorithm is considerably quick.

3 Matrices generation

In order to create encryption-decryption binary matrices pair, which are inverse to each other, the creation process begins with an identity matrix of a certain size. Example of an identity matrix of size 3x3:

$$M_i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

3.1 Encryption matrix generation

Creating an encryption matrix (public key) is based on performing multiple *elementary operations* on a matrix. There are two *elementary operations* used in the algorithm.

1. Swapping given two rows. For example swapping rows 1 and 2:

$$M_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \longrightarrow \quad M_2 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

2. Adding every corresponding element of given two rows in modulo 2 and storing the results on an indicated row. For example adding every corresponding element in modulo 2 of rows 1 and 3 and storing result on row 1:

$$M_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \longrightarrow \quad M_2 = \begin{bmatrix} 1+0 \equiv 1\,(mod\,2) & 0+0 \equiv 0\,(mod\,2) & 0+1 \equiv 1\,(mod\,2) \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \longrightarrow \quad M_2 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

3.2 Decryption matrix generation

Creating a decryption matrix (private key) is as simple as performing exactly the same *elementary operations* that have been performed on the encryption matrix but in the reversed order.

4 Encryption

In order to encrypt a binary message using an encryption matrix of size n x n, the message must be cut into vectors, each of n elements. If the length of the message is not a multiple of n, the last vector must be complemented to the size of n with zeros. After encryption, all vectors are linked together to form the encrypted message.

4.1 Vector encryption

Each element in the vector is corresponding to each row of matrix in the following way:

$$M_{EN} = \begin{bmatrix} m_{1,1} & m_{1,2} & \cdots & m_{1,n} \\ m_{2,1} & m_{2,2} & \cdots & m_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n,1} & m_{n,2} & \cdots & m_{n,n} \end{bmatrix} \quad V = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

Encrypted vector is a result of matrix M_{EN} and vector V multiplication in modulo 2:

$$V_{EN} = \left(\begin{bmatrix} m_{1,1} & m_{1,2} & \cdots & m_{1,n} \\ m_{2,1} & m_{2,2} & \cdots & m_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n,1} & m_{n,2} & \cdots & m_{n,n} \end{bmatrix} \cdot \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} \right) mod\, 2 = \begin{bmatrix} m_{1,1} \cdot v_1 + m_{1,2} \cdot v_2 + \cdots + m_{1,n} \cdot v_n \\ m_{2,1} \cdot v_1 + m_{2,2} \cdot v_2 + \cdots + m_{2,n} \cdot v_n \\ \vdots \\ m_{n,1} \cdot v_1 + m_{n,2} \cdot v_2 + \cdots + m_{n,n} \cdot v_n \end{bmatrix} mod\, 2$$

5 Decryption

Decryption algorithm works exactly as the encryption algorithm but instead of using the encryption matrix M_{EN} it uses the decryption matrix M_{DE}. After decryption, all vectors are linked together to form the decrypted message.

5.1 Vector decryption

Each element in the vector is corresponding to each row of matrix in the following way:

$$M_{DE} = \begin{bmatrix} m_{1,1} & m_{1,2} & \cdots & m_{1,n} \\ m_{2,1} & m_{2,2} & \cdots & m_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n,1} & m_{n,2} & \cdots & m_{n,n} \end{bmatrix} \quad V = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

Decrypted vector is a result of matrix M_{DE} and vector V multiplication in modulo 2:

$$V_{DE} = \left(\begin{bmatrix} m_{1,1} & m_{1,2} & \cdots & m_{1,n} \\ m_{2,1} & m_{2,2} & \cdots & m_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n,1} & m_{n,2} & \cdots & m_{n,n} \end{bmatrix} \cdot \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} \right) mod\, 2 = \begin{bmatrix} m_{1,1} \cdot v_1 + m_{1,2} \cdot v_2 + \cdots + m_{1,n} \cdot v_n \\ m_{2,1} \cdot v_1 + m_{2,2} \cdot v_2 + \cdots + m_{2,n} \cdot v_n \\ \vdots \\ m_{n,1} \cdot v_1 + m_{n,2} \cdot v_2 + \cdots + m_{n,n} \cdot v_n \end{bmatrix} mod\, 2$$

6 Encryption & decryption example

Let message be as follows:

$$m = 101011$$

Lets set a pair of $3x3$ encryption-decryption matrices, which are inverse to each other:

$$M_{EN} = {M_{DE}}^{-1}$$

$$M_{EN} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix} \quad M_{DE} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

6.1 Encryption process

Firstly, lets divide it into vectors V_i, each of length 3.

$$V_1 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \quad V_2 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

Lets encrypt V_1 by multiplying matrix M_{EN} by vector V_1.

$$V_{1EN} = \left(\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \right) mod\, 2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

$$V_{1EN} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

Now, lets encrypt V_2 by multiplying matrix M_{EN} by vector V_2.

$$V_{2EN} = \left(\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \right) mod\, 2 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$V_{2EN} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

So the encrypted message (ciphertext) c is a sequence of V_{1EN} and V_{2EN} elements.

$$c = 010100$$

6.2 Decryption process

The ciphertext from the previous example is:

$$c = 010100$$

In order to decrypt the ciphertext, lets divide it into vectors V_i, each of length 3.

$$V_1 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad V_2 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Lets decrypt V_1 by multiplying matrix M_{DE} by vector V_1.

$$V_{1DE} = \left(\begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \right) mod\, 2 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

$$V_{1DE} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

Now, lets decrypt V_2 by multiplying matrix M_{DE} by vector V_2.

$$V_{2DE} = \left(\begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \right) mod\, 2 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

$$V_{2DE} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

So the decrypted message m is a sequence of V_{1DE} and V_{2DE} elements

$$m = 101011,$$

which is the same as the original message in the beginning of the section 6, showing that the algorithm works.

References

[1] Andrzej Szablewski; Radosław Peszkowski; Mateusz Janus; Jakub Mazur. "Od problemu żarówek do nowego systemu szyfrowania (Polish) [From the lightbulb problem to the new encryption system]". 2016.

[2] Ran Raz. "On the Complexity of Matrix Product". In: *Proceedings of the Thiry-Fourth Annual ACM Symposium on Theory of Computing*. STOC '02. Montreal, Quebec, Canada: Association for Computing Machinery, 2002, pp. 144–151. ISBN: 1581134959. DOI: 10.1145/509907.509932. URL: https://doi.org/10.1145/509907.509932.